



Functions Guide

Table of Contents

I. Introduction	4
II. Function Descriptions.....	5
1) Registration and Other Functions	5
PCEX_Register(ServerName ; ServerPort ; ServerPage ; LicenseID).....	5
PCEX_GetOperatingMode.....	6
PCEX_Version(Type)	6
PCEX_GetLastError(Type)	7
PCEX_Authenticate(Email ; optUsername ; optPassword ; optDomain ; optAutodiscoverURL ; optExchangeVers).....	8
PCEX_BeginSession(AppID ; TenantID).....	9
PCEX_Authorize(AuthString)	10
PCEX_SaveSessionInfo	11
PCEX_LoadSessionInfo(SessionString).....	12
2) Record Specific Functions	13
PCEX_DeleteAllRecords(bExcludePrivate ; bExcludeDistList)	13
PCEX_DeleteCurrentRecord(bPermanent).....	13
PCEX_DeleteRecord(EntryID ; bPermanent)	14
PCEX_EmptyAllDeletedItems	15
PCEX_FilterByLastModified(Timestamp).....	16
PCEX_GetFieldData(FieldName).....	16
PCEX_GetContactImage	17
PCEX_GetCustomFieldData(FieldName ; FieldType).....	18
PCEX_MoveRecord(FolderPath).....	19
PCEX_NewRecord(optType)	20
PCEX_SaveRecord(optionalParam)	21
PCEX_SetCustomFieldData(FieldName ; FieldData ; FieldType)	22
PCEX_SaveRecordAsEML(Path)	23
PCEX_SendEmail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; optUseHTML).....	24
PCEX_SetContactImage(Image).....	25
PCEX_SetFieldData(FieldName ; FieldData)	26
PCEX_VerifyFieldData(FieldName ; VerifyData)	26
3) Attachment Specific Functions	27
PCEX_GetAttachmentCount.....	27

PCEX_GetAttachmentName(Index).....	27
PCEX_GetAttachmentSize(Index).....	28
PCEX_GetAttachmentCID(Index).....	29
PCEX_SaveAttachment(FilePath ; Index).....	30
PCEX_AddAttachment(ContainerOrFilePath ; optFileName ; blsContainer)	30
4) Folder Specific Functions	31
PCEX_GetCurrentFolderName	31
PCEX_GetDefaultFolderName(ItemType)	32
PCEX_GetFirstFolder	33
PCEX_GetFirstRecord	33
PCEX_GetFolderID.....	34
PCEX_GetFolderType.....	34
PCEX_GetNextFolder.....	34
PCEX_GetNextRecord.....	35
PCEX_GetRecordCount	35
PCEX_NewFolder(Name ; Type)	36
PCEX_OpenFolder(FolderPath)	37
PCEX_OpenRecord(EntryID)	37
PCEX_OpenMailbox(MailboxName).....	38
PCEX_OpenSharedFolder(ContactName ; Folder)	39
III. Available Record Fields	40
Available Common Fields	40
Available Contact Fields	41
Available Mail Fields.....	44
Available Calendar/Appointment Fields	45
Available Task Fields.....	46
Available Note Fields.....	47
V. Contact Us.....	48

I. Introduction

Description:

The Exchange Manipulator SE (“Server Edition”) plug-in is a powerful tool used to exchange data between FileMaker Pro® and one or more Microsoft® Exchange mailboxes. These operations are accomplished using FileMaker function calls from within FileMaker calculations. These calculations are generally determined from within FileMaker “Set Field”, “Set Variable” or “If” script steps. This document describes all available plug-in functions and fields. Please see the accompanying “Developer’s Guide” for a list of the integration steps, features and error handling.

Intended Audience:

FileMaker developers or persons who have knowledge of FileMaker scripting, calculations and relationships as proper use of the plug-in requires that FileMaker integration scripts be created in your FileMaker solution.

Successful Integration Practices:

- 1) Read the Developer’s Guide
- 2) Read the Functions Guide
- 3) Familiarize yourself with Microsoft Exchange
- 4) Review our FileMaker demo file
- 5) Watch our video tutorials: <https://help.productivecomputing.com/help/exchange-manipulator-se>

II. Function Descriptions

This section describes the functions that are available with the Exchange Manipulator SE plug-in.

1) Registration and Other Functions

PCEX_Register(ServerName ; ServerPort ; ServerPage ; LicenseID)

Purpose:

This function registers the plug-in with Productive Computing's registration servers. You must call the Register function once before using the plug-in.

In server implementations, the Register function must be called once per virtual "session", such as when calling a script with Perform Script on Server.

Dependencies:

None.

Parameters:

Parameter Name	Purpose	Values
ServerName	The name of the server to register with	"licensing.productivecomputing.com"
ServerPort	The port of the server to register with	"80"
ServerPage	Registration page path	"/PCIReg/pcireg.php"
LicenseID	The license ID to register with	"DEMO-EMS" for demo evaluation, or your purchased license ID

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_GetOperatingMode

Purpose:

This function returns the current operating mode of the plug-in.

Dependencies:

None.

Parameters:

None

Return Values:

"UNREGISTERED", "EXPIRED", "DEMO", or "LIVE".

Notes & Examples:

Valid results are:

- UNREGISTERED: The plug-in has not been registered with the Productive Computing, Inc., licensing server.
- EXPIRED: The plug-in is running in DEMO mode, but the Demo time has expired. Restart FileMaker to continue evaluation.
- DEMO: The plug-in is currently running in DEMO mode and will function as normal during a 2-hour FileMaker session window. After that, it will become EXPIRED.
- LIVE: The plug-in is currently running in LIVE mode and will function as normal and will never expire.

PCEX_Version(Type)

Purpose:

This function returns the version of the plug-in, depending on the Type parameter.

Dependencies:

None.

Parameters:

Parameter Name	Purpose	Values
Type	The type of version info to return	"Long" returns the full name with version number. "Short" returns just the version number.

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

If the plug-in's version is 1.0.0.0:

PCEX_Version("Long") would return "Exchange Manipulator SE v1.0.0.0"

PCEX_Version("Short") would return "1.0.0.0"

PCEX_GetLastError(Type)**Purpose:**

This function returns the last error encountered by the plug-in. If specified, the Type determines what kind of error result to return.

Dependencies:

None.

Parameters:

Parameter Name	Purpose	Values	Default Value
Type	The type of error to return.	"Number" will return the error code. "Text", "", or no parameter will return the error text.	"Text"

Return Values:

The error Number or Text, or "!!ERROR!!"

PCEX_Authenticate(Email ; optUsername ; optPassword ; optDomain ; optAutodiscoverURL ; optExchangeVers)

***DEPRECATED**

Purpose:

This function authenticates the plug-in with an Exchange service belonging to the Email address. If provided, the username, password, and domain fields will clarify the specific Exchange service to communicate with. If left blank, the process will authenticate using the credentials of the currently-logged in Windows user. The optAutodiscoverURL allows for the developer to specify the specific Autodiscover URL instead of letting the plug-in automatically determine it; this is useful for more advanced Exchange server deployments. The optExchangeVers function allows the developer to specify which version of Exchange server to communicate with, and changing the Exchange version may also disable or enable certain features the plug-in can communicate with.

Dependencies:

None

Parameters:

Parameter Name	Purpose
Email	The email address of the Exchange user to authenticate as. This is the same as the user's Exchange mailbox name.
optUsername	(Optional) The user name of the Exchange user to authenticate as.
optPassword	(Optional) The password of the Exchange user to authenticate as.
optDomain	(Optional) The Exchange domain of the Exchange user to authenticate as.
optAutodiscoverURL	(Optional) The URL reference for the Exchange server's Autodiscover service.
optExchangeVers	(Optional) The version of Exchange server to communicate with.

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

As of version 3.0.0.0, the PCEX_Authenticate function has been deprecated, as this method was primarily used to communicate with on-premise Exchange servers using Basic Authentication. Basic authentication has been deprecated by Microsoft and connections to Exchange servers through basic authentication will be prevented in the near future.

Please make use of the PCEX_BeginSession / PCEX_Authorize functions to ensure that your integration with Microsoft Exchange continues uninterrupted, and review the Developer's Guide for instructions on how to implement OAuth 2.0 authentication.

support older server versions in the optExchangeVers parameter. For a list of valid Exchange server parameters, please refer to the "Additional Fields" section.

It is advised to always specify the username, password, and domain name of the account you wish to authenticate to when calling this function, as that will ensure that the account is always the correct one, especially when run from different environments. If using the plug-in in a context that doesn't have a default domain name (such as with a local non-domain user account, or from a remote desktop client on another network), the optional parameters become required.

The email address and authentication credentials dictate what permissions the plug-in has available if talking to other Exchange mailboxes using PCEX_OpenMailbox. These permissions are managed on a per-Mailbox level. Functions that may be affected by permissions (such as PCEX_NewRecord, PCEX_SetFieldData, or PCEX_SaveRecord) will report an error if there are insufficient permissions to carry out their tasks.

If in an Exchange-enabled environment that uses a Domain, the Domain for the mailbox can potentially be the same as the domain of the Windows user account. If not specified, the plug-in will use the default domain associated with the Email parameter (which is the segment after the “@” in the email address; e.g. if the email is Test.Testerson@testcompany.com, the domain would by default be testcompany.com).

PCEX_BeginSession(AppID ; TenantID)

Purpose:

This function begins an authenticated session with Exchange Online’s servers, using the provided AppID and TenantID to identify the Exchange service connection. This will launch an instance of the default Windows web browser and prompt the user to log in with their Microsoft account. If the account is allowed to connect, the page will be redirected to the redirect URL configured with the Exchange Manipulator’s app entry on the organization’s Azure AD portal. This redirect URL should provide critical access information in order to complete the authentication process.

Dependencies:

None

Parameters:

Parameter Name	Purpose
AppID	The application ID, in the format of a GUID
TenantID	The application’s Tenant ID, in the format of a GUID

Return Values:

0 for success, otherwise “!!ERROR!!”

Notes and Examples:

Please ensure to follow the instructions in the Developer’s Guide on configuring the app registration for Exchange Manipulator on your Exchange Online’s Azure AD instance.

After this function is called, the function “PCEX_Authorize” should be called. The parameter for the Authorize function should be the encoded string provided by the final web page after the user logs in on the website.

PCEX_Authorize(AuthString)

Purpose:

This function finalizes the authentication process by accepting an encoded AuthString, which was generated after the user signs into the web page brought up by PCEX_BeginSession. The AuthString is decoded and processed, and the plug-in validates that the connection is good. If successful, the plug-n will now be considered 'authenticated' as the user that was chosen when logging in to Microsoft, and the function will return that user's email address.

Dependencies:

None

Parameters:

Parameter Name	Purpose
AuthString	Encoded string containing authentication information for the

Return Values:

The authenticated user's email address for success, otherwise "!!ERROR!!"

Notes and Examples:

Please ensure to follow the instructions in the Developer's Guide on configuring the app registration for Exchange Manipulator on your Exchange Online's Azure AD instance.

This function will finalize the authentication process started with "PCEX_BeginSession". This function is NOT compatible with the "PCEX_Authenticate" function, which is used for authenticating with "Basic" authentication for on-premise Exchange servers.

PCEX_SaveSessionInfo

Purpose:

This function saves the current authentication state of the plug-in into an encoded string, allowing transfer of connection information between FileMaker sessions, and especially between Client and Server instances of the plug-in. It is important to note that this session information is specifically for authenticated sessions created via the PCEX_BeginSession / PCEX_Authorize functions, and NOT for sessions authenticated via PCEX_Authenticate.

Dependencies:

None

Parameters:

None

Return Values:

The encoded session string for success, otherwise “!!ERROR!!”

Notes and Examples:

Please ensure to follow the instructions in the Developer’s Guide on configuring the app registration for Exchange Manipulator on your Exchange Online’s Azure AD instance.

This function will finalize the authentication process started with “PCEX_BeginSession”. This function is NOT compatible with the “PCEX_Authenticate” function, which is used for authenticating with “Basic” authentication for on-premise Exchange servers.

PCEX_LoadSessionInfo(SessionString)

Purpose:

This function accepts an encoded session string generated by PCEX_SaveSessionInfo, decodes the string, parses the contents and replicates the session information consistent with a fully authenticated connection with Exchange Online. During the process of loading session information, the access token will be checked in case it needs to be refreshed; if so, the token will be automatically refreshed, and the new access token is stored in the plug-in's memory. With this in mind, we advise that after calling PCEX_LoadSessionInfo successfully, developers should immediately call PCEX_SaveSessionInfo, so that if the access token is refreshed, it will be preserved for further use and help mitigate authentication issues down the line.

Dependencies:

None

Parameters:

Parameter Name	Purpose
SessionString	Encoded session string generated by a prior call to PCEX_SaveSessionInfo

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes and Examples:

Please ensure to follow the instructions in the Developer's Guide on configuring the app registration for Exchange Manipulator on your Exchange Online's Azure AD instance.

This function will finalize the authentication process started with "PCEX_BeginSession". This function is NOT compatible with the "PCEX_Authenticate" function, which is used for authenticating with "Basic" authentication for on-premise Exchange servers.

2) Record Specific Functions

PCEX_DeleteAllRecords(bExcludePrivate ; bExcludeDistList)

Purpose:

This function will delete all records in the currently-opened folder. If specified as true, bExcludePrivate will exclude any records marked "Private", and bExcludeDistList will exclude any records that are part of a distribution list (contacts only).

The deletion of records is PERMANENT. This cannot be undone.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

Parameter Name	Purpose	Values	Default Value
bExcludePrivate	If true, the delete operation will not delete any Private records.	True or False	false
bExcludeDistList	If true, the delete operation will not delete any Contact records that are part of a distribution list.	True or False	false

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_DeleteCurrentRecord(bPermanent)

Purpose:

This function deletes the currently-opened record. If specified as true, the operation will be permanent and cannot be undone. Otherwise, the plug-in will move the record into the open mailbox's Deleted Items folder.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose	Values	Default Value
bPermanent	If true, the delete operation will be permanent. If false or not specified, the current record will be moved to "Deleted Items".	True or False	false

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Permanent deletions will completely remove the deleted item(s) and cannot be undone. Use with caution.

PCEX_DeleteRecord(EntryID ; bPermanent)**Purpose:**

This function deletes the record with the specified Entry ID in any folder. If specified, the bPermanent parameter controls whether the deletion is permanent, or if the record will be moved to the Deleted Items folder.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
Entry ID	The unique identifier of the record to be deleted. This ID should be an Item-type Entry ID (cannot delete folders).
bPermanent	If true, the delete operation will be permanent. Otherwise, the record will be moved to the "Deleted Items" folder.

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Permanent deletions will completely remove the deleted item(s) and cannot be undone. Use with caution.

PCEX_EmptyAllDeletedItems

Purpose:

This function will empty the Deleted Items folder, clearing out all items and permanently deleting them. All items in the Deleted Items folder will be lost and cannot be recovered.

Dependencies:

Must have authenticated to Exchange.

Parameters:

None

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Permanent deletions will completely remove the deleted item(s) and cannot be undone. Use with caution.

PCEX_FilterByLastModified(Timestamp)

Purpose:

This function will apply a timestamp filter to any records belonging to the currently-opened folder. When accessing records within a folder after applying the filter, only those records that were modified on or after the timestamp will be available for use.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
Timestamp	A valid timestamp (FileMaker-format) with which to apply the filter

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

When working with filters, the general recommendation is to open the desired folder, apply the filter, and then check the record count. If the record count is greater than 0, then there are records to be processed; otherwise, no records matched the filter criteria.

PCEX_GetFieldData(FieldName)

Purpose:

This function returns the field data for the currently-opened record belonging to the provided field name.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose
FieldName	The name of the field

Return Values:

The value of the desired field, or "!!ERROR!!"

PCEX_GetContactImage

Purpose:

This function returns the binary container data representing the Contact Picture assigned to the currently-opened Contact record. If no picture has been set, the result will be blank.

Dependencies:

Must have authenticated to Exchange. Must have opened a Contact record.

Parameters:

None

Return Values:

Binary container data for the image, blank, or "!!ERROR!!".

PCEX_GetCustomFieldData(FieldName ; FieldType)

Purpose:

Extracts the value contained in the specified custom field that has the provided type. The field must exist in the record that is currently opened.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	Name of the custom field		
FieldType	Data type of the custom field	See acceptable values.	"Text"

Return Values:

The string representation of the field value, or "!!ERROR!!"

Notes & Examples:

Acceptable FieldType values:

- "Currency"
- "Date"
- "Time"
- "DateTime"
- "Duration"
- "Keywords"
- "Number"
- "Percent"
- "Text"
- "YesNo"
- "Integer"

PCEX_MoveRecord(FolderPath)

Purpose:

This function moves the currently-opened record to the specified folder path.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose
FolderPath	The path of the folder where the record will be moved to.

Return Values:

The new Entry ID of the moved record, or "!!ERROR!!"

Notes & Examples:

This function can only move records within the same Mailbox. It cannot move records across Mailboxes.

As of version 3.0.0.0, the Exchange Manipulator plug-in has been updated to communicate via the Microsoft Graph API. The Graph API does not support the moving of Contacts or Calendar Events; PCEX_MoveRecord therefore only supports the movement of Mail records between folders. This is a known issue and has been reported to Microsoft for future enhancement to the Graph API.

PCEX_NewRecord(optType)

Purpose:

This function creates a new record within the currently-opened folder.

If specified, the optType parameter will dictate what type of record to create. Otherwise, it will take the type belonging to the currently-opened folder.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

Parameter Name	Purpose	Values
optType	(Optional) The type of record to create.	"Contact", "Mail", "Event", "Meeting", "Appointment", "Task".

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Available Types:

- Contact
- Mail
- Event
- Meeting
- Appointment

PCEX_SaveRecord(optionalParam)

Purpose:

This function saves the currently-opened record, committing any changes that have been made. Based on the presence and value of the optionalParam parameter, the plug-in will perform different actions with the saved record.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose	Values
optionalParam	The optional parameter to set the behavior of how to save the record.	If left blank, this will simply save and commit the record. Refer to the Notes for specifics on what each value does.

Return Values:

The Entry ID of the record, or "!!ERROR!!"

Notes & Examples:

Acceptable parameters:

- "DontSend": If saving a Mail record, the email will be saved and placed in the opened mailbox's Drafts folder. The user can then edit the message at a later time, whether through an email client or the plug-in, and send it later.
- "Send" or "Appointment": If saving an Event record, the event will be saved and will send out a meeting notice to any attendees. This is required if submitting a meeting cancellation to the attendees.
- "ApptResponse": If saving an Event record, the event will be saved and a meeting response notification will be submitted to the meeting's organizer. This notification can be an Accept, Decline, or Tentative Accept notification.

PCEX_SetCustomFieldData(FieldName ; FieldData ; FieldType)

Purpose:

This function creates and sets a custom field within the currently-opened record and assigns it a given value. The field is created with the specified data type.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose	Values	Default Value
FieldName	Name of the custom field		
FieldData	Data to assign to the custom field		
FieldType	Data type of the custom field	See acceptable type values	"Text"

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Acceptable FieldType values:

- "Currency"
- "Date"
- "Time"
- "DateTime"
- "Duration"
- "Keywords"
- "Number"
- "Percent"
- "Text"
- "YesNo"
- "Integer"

PCEX_SaveRecordAsEML(Path)

Purpose:

This function saves the current record and all of its corresponding information to a .EML file, and stores that file at the given Path. The .EML file is compatible with most email applications, and is a good way to preserve copies of mail records.

Dependencies:

Must have authenticated to Exchange. Must have opened a Mail-type record.

Parameters:

Parameter Name	Purpose
Path	A Windows-format file path

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

If using this function in a server-side capacity, it is advised to provide a path to a shared file server or network drive with a mapped drive letter that is available from the server. The server-side demo has an example of inserting the saved EML file back into the demo file after saving.

PCEX_SendEmail(To ; Cc ; Bcc ; Subject ; Body ; Attachments ; optUseHTML)

Purpose:

This function sends an email using the specified parameters, and is otherwise known as the "Short" method of sending email. This function does not return an Entry ID.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose	Values	Default Value
To	The email address(es) that will be the To recipients	Multiple email addresses are separated by semicolon (;) characters.	
Cc	The email address(es) that will be the Cc recipients	Multiple email addresses are separated by semicolon (;) characters.	
Bcc	The email address(es) that will be the Bcc recipients	Multiple email addresses are separated by semicolon (;) characters.	
Subject	The subject of the email message		
Body	The body of the email message		
Attachment	A Windows-format file path referencing one or more files to attach to the email	Multiple attachments are separated by comma (,) characters, and each must be a fully-qualified Windows style file path.	
optUseHTML	(Optional) Flag to set the email to send as HTML or Plain Text.	"True" = HTML format. "False" = Plain Text format.	False

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

The Attachments parameter is a list of attachments separated by commas. Each attachment entry must be a full, Windows-format path. See the example below:

```
PCEX_SendMail(
"toPerson@email.com" ;
"ccPerson@email.com" ;
"bccPerson@email.com" ;
"Test Subject" ;
"This is a test body. The body would go here." ;
"C:\\Temp\\PictureA.jpg,C:\\Temp\\DocumentB.docx,E:\\External Files\\RemoteFile.txt" ;
True )
```

This function call will send an email titled "Test Subject" with the body of "This is a test body. The body would go here." to the toPerson email address, cc'ing the ccPerson email address, and sending a blind carbon copy back to the bccPerson email address. This email would be sent as an HTML-format email, and would include three attachments.

PCEX_SetContactImage(Image)**Purpose:**

This function assigns the specified Image container data to the currently-opened contact record's Contact Picture reference.

Dependencies:

Must have authenticated to Exchange. Must have opened a Contact record.

Parameters:

Parameter Name	Purpose	Values
Image	Binary container data representing the image.	Image file must be of type .bmp, .jpg/.jpeg, .gif, or .png.

Return Values:

0 for success, otherwise "!!ERROR!!".

Notes & Examples:

This image data must be present within the container field; storing the data by reference will not properly set the contact picture.

Contact pictures are mostly for use with other third party applications that reference the same Exchange Contact data, such as Outlook, Apple Contacts, or other similar apps.

PCEX_SetFieldData(FieldName ; FieldData)

Purpose:

This function sets the field at the specified Field Name with the provided data.

Dependencies:

Must have authenticated to Exchange. Must have opened or created a valid record.

Parameters:

Parameter Name	Purpose
FieldName	Name of the field
FieldData	The data to set for the field

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_VerifyFieldData(FieldName ; VerifyData)

Purpose:

This function compares the existing value of the field specified by FieldName with the provided VerifyData.

Dependencies:

Must have authenticated to Exchange. Must have opened a record.

Parameters:

Parameter Name	Purpose
FieldName	Name of the field to verify
VerifyData	Data to compare the field's contents with

Return Values:

0 if the data values are the same, otherwise "!!ERROR!!"

3) Attachment Specific Functions

PCEX_GetAttachmentCount

Purpose:

This function returns the total number of attachments for the currently-opened Mail record.

Dependencies:

Must have authenticated to Exchange. Must have opened a Mail-type record.

Parameters:

None

Return Values:

The count of attachments, or "!!ERROR!!"

PCEX_GetAttachmentName(Index)

Purpose:

This function gets the name of the attachment linked to the currently-opened Mail record at the specified index.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
Index	The Index of the attachment to get the name of. This index is 1-based; the first attachment is at index 1.

Return Values:

The name of the attachment, or "!!ERROR!!"

PCEX_GetAttachmentSize(Index)

Purpose:

This function gets the size of the attachment, in bytes, at the specified Index.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
Index	The index of the attachment to get the size of. This index is 1-based; the first attachment is at index 1.

Return Values:

The size of the attachment in bytes, or "!!ERROR!!"

PCEX_GetAttachmentCID(Index)

Purpose:

This function returns the Content ID (or "CID") of the attachment at the specified index. The Content ID of an attachment is set by the Exchange mail protocol when an image or file is embedded within the Body of an email message. Attachments that are not embedded will have a blank Content ID.

Dependencies:

Must have authenticated to Exchange. Must have opened a Mail-type record.

Parameters:

Parameter Name	Purpose
Index	The Index of the attachment to get the name of. This index is 1-based; the first attachment is at index 1.

Return Values:

The CID of the attachment, or "!!ERROR!!"

Notes and Examples:

Embedded attachments are usually found after attached attachments in the attachment list.

Example:

Suppose an email is received that contains an attachment entitled "Some PDF.pdf", and contains an image embedded within the Body titled "Smile.png".

When getting the HTML Body field data of the email, there will be an reference tag that references Smile.png within the body, and contains some generated characters identifying that embedded image (e.g. "Smile.png@AB8FE224").

When calling PCEX_GetAttachmentCID and specifying the index of that Smile.png file, the result will be "Smile.png@AB8FE224".

PCEX_SaveAttachment(FilePath ; Index)

Purpose:

This function saves the attachment located at the specified index to a provided file path.

Dependencies:

Must have authenticated to Exchange. Must have opened a Mail-type record.

Parameters:

Parameter Name	Purpose	Values
FilePath	The Windows-format path for the attachment	
Index	The index of the attachment to save	Indices are 1-based. The first attachment will always be at index 1.

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_AddAttachment(ContainerOrFilePath ; optFileName ; blsContainer)

Purpose:

This function adds an attachment to the currently-opened Mail record. This attachment can come from a container field or using a file path. If the attachment is coming from a container field, the parameter "blsContainer" must be set to 1 or True, you must provide a file name, and the file being attached must be embedded. If the attachment is from a file path, blsContainer must be 0 or False and the file name can be blank; if it is blank, it will be the file name in the path, otherwise the file will be named the provided file name.

Dependencies:

Must have an open Mail record, whether through PCEX_NewRecord or PCEX_OpenRecord. Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
ContainerOrFilePath	The FileMaker container field or a file path pointing to the file to attach.
optFileName	File name to attach the file as. Optional if a file path; required if a container.
blsContainer	Boolean. Indicates the ContainerOrFilePath parameter is a container or not.

Return Values:

0 for success, otherwise "!!ERROR!!"

4) Folder Specific Functions

PCEX_GetCurrentFolderName

Purpose:

This function returns the currently-opened folder's name.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

On success, the name of the current folder. Otherwise, "!!ERROR!!".

PCEX_GetDefaultFolderName(ItemType)

Purpose:

This function gets the name of the default folder associated with the provided Item Type.

Dependencies:

Must be authenticated to Exchange.

Parameters:

Parameter Name	Purpose
ItemType	The type of folder

Return Values:

The default folder name of the item type, or "!!ERROR!!"

Notes & Examples:

Valid Item Types:

- Deleted Items
- Outbox
- Sent Mail
- Inbox
- Calendar
- Contacts
- Journal
- Notes
- Tasks
- Drafts
- All Public Folders
- Junk
- Root

PCEX_GetFirstFolder

Purpose:

This function gets the first subfolder belonging to the currently-opened folder or Mailbox.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The path of the first subfolder, otherwise "!!ERROR!!"

PCEX_GetFirstRecord

Purpose:

This function returns the first record present in the currently-opened folder.

If a filter has been applied, the record will be the first record that matches the filter's criteria.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The entry ID of the first record, or "!!ERROR!!"

PCEX_GetFolderID

Purpose:

This function returns the Entry ID of the currently-opened Folder.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The folder's Entry ID, or "!!ERROR!!"

PCEX_GetFolderType

Purpose:

This function returns the type of the currently-opened folder.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The folder's type, or "!!ERROR!!"

PCEX_GetNextFolder

Purpose:

This function gets the next subfolder belonging to the currently-opened folder or Mailbox. If no other folders are available, this function returns "End".

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The path of the next folder, "End" if there are no more folders to get, or "!!ERROR!!" if there was some error.

PCEX_GetNextRecord**Purpose:**

This function gets the next record found under the currently-opened folder. If no other record is available, the function will return "End".

If a filter has been applied, the record will be the next record that matches the filter's criteria.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The Entry ID of the next record, "End" if there are no more records left, or "!!ERROR!!" if an error was encountered.

PCEX_GetRecordCount**Purpose:**

This function returns the count of records belonging to the currently-opened folder. This function is especially useful when checking to see if there are records to process after calling PCEX_FilterByLastModified.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder.

Parameters:

None

Return Values:

The count of records available, otherwise "!!ERROR!!"

PCEX_NewFolder(Name ; Type)

Purpose:

This function creates a new folder with the specified name and type, and save it to the currently-opened folder as a subfolder.

If the Type is not provided, the new folder will take the type of the currently-opened folder type.

After creating the folder successfully, you must then open that folder in order to create items within it.

Dependencies:

Must have authenticated to Exchange. Must have opened a folder or mailbox.

Parameters:

Parameter Name	Purpose
Name	The name of the folder to create.
Type	The type of the folder to create.

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

Available Types:

- Contact
- Calendar
- Mail

PCEX_OpenFolder(FolderPath)

Purpose:

This function opens the folder at the specified folder path.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
FolderPath	The full path of the folder to open.

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_OpenRecord(EntryID)

Purpose:

This function opens a record with the specific Entry ID. This record could exist in any folder.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose
EntryID	The Entry ID of the record to open.

Return Values:

0 for success, otherwise "!!ERROR!!"

PCEX_OpenMailbox(MailboxName)

Purpose:

This function opens the mailbox belonging to the specified MailboxName parameter.

See notes for Mailbox-related details.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose	Values
MailboxName	The name of the mailbox to open	Mailbox names are the same as the Exchange email account. Example: Test.Testerson@testcompany.com

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

If the MailboxName refers to the same mailbox as the authenticated user, the plug-in will have full control over all elements of the mailbox (each user has Owner-level control over their own mailbox).

If the MailboxName refers to a different mailbox than the authenticated user, the plug-in will only have permissions and control over the elements of that mailbox according to the permission set in Exchange associated with the authenticated user.

For example:

User A has Owner control over User A's Mailbox.

User B has Owner control over User B's Mailbox.

User B also has Contributor control over User A's Mailbox.

When authenticating as User B and opening User A's Mailbox, the plug-in will have Contributor control over User A's Mailbox records; that is, the plug-in will only be able to create records, but cannot edit or delete items, and cannot create, edit, or delete folders.

PCEX_OpenSharedFolder(ContactName ; Folder)

Purpose:

This function opens the shared folder specified by the Folder number that has been shared with the authenticated account by the user's Contact Name. The plug-in has the same access permissions on the shared folder as has been granted by the sharing contact.

Dependencies:

Must have authenticated to Exchange.

Parameters:

Parameter Name	Purpose	Values
ContactName	The name of the contact sharing the folder.	
Folder	The folder number representing the shared folder.	See the values in the Notes & Examples section

Return Values:

0 for success, otherwise "!!ERROR!!"

Notes & Examples:

The Folder parameter is one of the following:

Inbox = 6

Calendar = 9

Contacts = 10

Journal = 11

Notes = 12

Tasks = 13

Drafts = 16

Conflicts = 19

SyncIssues = 20

LocalFailures = 21

ServerFailures = 22

Junk = 23

III. Available Record Fields

Exchange has various records or item types such as Contact, Task, Mail, Calendar, and Note items. Each item contains numerous fields, which have values that can be read or set. In order to read or “get” a field value, you must call the PCEX_GetFieldData function and specify the name of the field you would like to get. In order to write or “set” a field value, you must call the PCEX_SetFieldData function and specify the name and value of the field you would like to set.

The fields below are separated by their applicable item type, and indicate whether you can “get” or “set” the field. If there is a missing field that you would like to access, then please email a request to support@productivecomputing.com.

Available Common Fields

These fields are available for every record type.

Method	Field Name	Acceptable Values/Notes
<i>Get</i>	Created	Alternate Name: Create. Creation timestamp.
<i>Get</i>	ID	Alternate Name: Entry ID. Unique identifier of the Contact.
<i>Get</i>	Form Name	Same as Message Class. It’s advised not to set this field; unexpected behavior may result from doing so.
<i>Get</i>	Message Class	Same as Form Name. It’s advised not to set this field; unexpected behavior may result from doing so.
<i>Get</i>	Modified	Alternate Name: Modify. Modification timestamp.

Available Contact Fields

Contact items represent contacts in a Contacts folder.

Method	Field Name	Acceptable Values/Notes
<i>Get/Set</i>	Account	
<i>Get/Set</i>	Anniversary	Date
<i>Get/Set</i>	Assistant's Name	
<i>Get/Set</i>	Assistant's Phone	
<i>Get/Set</i>	Billing Information	
<i>Get/Set</i>	Birthday	Date
<i>Get/Set</i>	Body	This is used for storing notes on the Contact
<i>Get/Set</i>	Business Address	Entire address as a block
<i>Get/Set</i>	Business Address City	
<i>Get/Set</i>	Business Address Country	
<i>Get/Set</i>	Business Address PO Box	
<i>Get/Set</i>	Business Address Postal Code	
<i>Get/Set</i>	Business Address State	
<i>Get/Set</i>	Business Address Street	
<i>Get/Set</i>	Business Fax	
<i>Get/Set</i>	Business Home Page	
<i>Get/Set</i>	Business Phone	
<i>Get/Set</i>	Business Phone 2	
<i>Get/Set</i>	Callback	
<i>Get/Set</i>	Car Phone	
<i>Get/Set</i>	Categories	
<i>Get/Set</i>	Children	
<i>Get/Set</i>	Company	
<i>Get/Set</i>	Company Main Phone	
<i>Get/Set</i>	Computer Network Name	
<i>Get/Set</i>	Customer ID	
<i>Get/Set</i>	Department	
<i>Get/Set</i>	Display Name	
<i>Get/Set</i>	Email	
<i>Get/Set</i>	Email Address	
<i>Get/Set</i>	Email 2	
<i>Get/Set</i>	Email Address 2	
<i>Get/Set</i>	Email 3	
<i>Get/Set</i>	Email Address 3	
<i>Get/Set</i>	Email Address Type	
<i>Get/Set</i>	Email 2 Address Type	
<i>Get/Set</i>	Email 3 Address Type	
<i>Get/Set</i>	Email Display As	
<i>Get/Set</i>	Email 2 Display As	
<i>Get/Set</i>	Email 3 Display As	
<i>Get/Set</i>	File As	
<i>Get/Set</i>	First Name	
<i>Get/Set</i>	FTP Site	
<i>Get/Set</i>	Full Name	

<i>Get/Set</i>	Gender	Female, Male, Unspecified.
<i>Get/Set</i>	Government ID Number	
<i>Get/Set</i>	Hobbies	
<i>Get/Set</i>	Home Address	Entire address as a block.
<i>Get/Set</i>	Home Address City	
<i>Get/Set</i>	Home Address Country	
<i>Get/Set</i>	Home Address PO Box	
<i>Get/Set</i>	Home Address Postal Code	
<i>Get/Set</i>	Home Address State	
<i>Get/Set</i>	Home Address Street	
<i>Get/Set</i>	Home Fax	
<i>Get/Set</i>	Home Phone	
<i>Get/Set</i>	Home Phone 2	
<i>Get/Set</i>	IM Address	
<i>Get/Set</i>	Importance	
<i>Get/Set</i>	Internet Free/Busy Status	
<i>Get/Set</i>	ISDN	
<i>Get/Set</i>	Job Title	
<i>Get/Set</i>	Journal	True or False
<i>Get/Set</i>	Language	
<i>Get/Set</i>	Last Name	
<i>Get/Set</i>	Location	
<i>Get/Set</i>	Mailing Address	Entire address as a block.
<i>Get/Set</i>	Manager's Name	
<i>Get/Set</i>	Middle Name	
<i>Get/Set</i>	Mileage	
<i>Get/Set</i>	Mobile Phone	
<i>Get/Set</i>	Nickname	
<i>Get/Set</i>	Office Location	
<i>Get/Set</i>	Organizational ID Number	
<i>Get/Set</i>	Other Address	Entire address as a block.
<i>Get/Set</i>	Other Address City	
<i>Get/Set</i>	Other Address Country	
<i>Get/Set</i>	Other Address PO Box	
<i>Get/Set</i>	Other Address Postal Code	
<i>Get/Set</i>	Other Address State	
<i>Get/Set</i>	Other Address Street	
<i>Get/Set</i>	Other Fax	
<i>Get/Set</i>	Other Phone	
<i>Get/Set</i>	Pager	
<i>Get/Set</i>	Personal Home Page	
<i>Get/Set</i>	Primary Phone	
<i>Get/Set</i>	Profession	
<i>Get/Set</i>	Radio Phone	
<i>Get/Set</i>	Referred By	
<i>Get/Set</i>	Selected Mailing Address	
<i>Get/Set</i>	Sensitivity	Confidential, Normal, Personal, Private
<i>Get/Set</i>	Spouse	

Get/Set	Subject	
Get/Set	Suffix	
Get/Set	Telex	
Get/Set	Title	
Get/Set	User Field 1	
Get/Set	User Field 2	
Get/Set	User Field 3	
Get/Set	User Field 4	
Get/Set	Webpage	

Note: When adding any Phone-type fields, the format of the string is as follows:

['+' & country code & " (" & area/city code & ") " & number & " x" & extension]

Example:

+1 (555) 555-5555 x123

- The country code is optional, but if present, it must be preceded by a "+" character and followed by a space.
- The city/area code is optional, but if included it is enclosed by parenthesis and followed by a space.
- The actual number is required.
- If an extension is included, it is prefixed with the "x" character.

Available Mail Fields

Mail items represent a mail message in an Inbox folder.

Method	Field Name	Acceptable Values/Notes
<i>Get/Set</i>	Bcc	
<i>Get/Set</i>	Billing Information	
<i>Get/Set</i>	Body	
<i>Get/Set</i>	Body Format	HTML, Plain, Rich Text
<i>Get/Set</i>	Categories	
<i>Get/Set</i>	Cc	
<i>Get</i>	Conversation Index	
<i>Get</i>	Conversation Topic	
<i>Get/Set</i>	Defer Until	Timestamp
<i>Get</i>	Expires	Timestamp
<i>Get/Set</i>	Flag Status	Complete, Marked, None
<i>Get/Set</i>	From	Alternate Names: From 1, From 2
<i>Get/Set</i>	Header	
<i>Get/Set</i>	HTML Body	
<i>Get/Set</i>	Importance	High, Low, Normal, or H, L, N
<i>Get/Set</i>	Originator Delivery Requested	True or False
<i>Get/Set</i>	Receipt Requested	True or False
<i>Get</i>	Received	Alternate Name: Received Time. Timestamp.
<i>Get/Set</i>	Remote Status	Marked for Copy, Marked for Delete, Marked for Download, None, Unmarked.
<i>Get/Set</i>	RTF Body	
<i>Get/Set</i>	Sensitivity	Confidential, Normal, Personal, Private
<i>Get</i>	Sent	Alternate Name: Sent On. Timestamp.
<i>Set</i>	Sent On Behalf Of	
<i>Get</i>	Size	
<i>Get/Set</i>	Subject	
<i>Get/Set</i>	To	
<i>Get/Set</i>	Unread	True or False
<i>Get/Set</i>	Voting Response	

Available Calendar/Appointment Fields

Appointment items represent a meeting, one-time appointment, recurring appointment, or meeting.

Method	Field Name	Acceptable Values/Notes
<i>Get/Set</i>	All Day Event	True or False
<i>Get/Set</i>	Billing Information	
<i>Get/Set</i>	Body	This is the “note” field
<i>Get/Set</i>	Busy Status	Busy, Free, Out Of Office, or Tentative
<i>Get/Set</i>	Categories	
<i>Get/Set</i>	Duration	Time in Minutes
<i>Get/Set</i>	End	Timestamp
<i>Get/Set</i>	Importance	High, Low, Normal (H, L, or N)
<i>Get</i>	Is Recurring	True or False
<i>Get/Set</i>	Location	
<i>Get/Set</i>	Meeting Response	Accepted, Declined, Tentative
<i>Get/Set</i>	Meeting Status	Meeting, Canceled, Received, NonMeeting
<i>Get</i>	Net Show URL	
<i>Get/Set</i>	Optional Attendees	Use semicolon between multiple attendees.
<i>Get</i>	Organizer	
<i>Get/Set</i>	Remind Beforehand	In Minutes
<i>Get/Set</i>	Reminder	True or False
<i>Get/Set</i>	Required Attendees	Use semicolon between multiple attendees.
<i>Get</i>	Resources	Use semicolon between multiple attendees.
<i>Get/Set</i>	Response Requested	True or False
<i>Get/Set</i>	Sensitivity	Confidential, Normal, Personal, Private.
<i>Get</i>	Size	
<i>Get/Set</i>	Start	Timestamp.
<i>Get/Set</i>	Subject	

Available Task Fields

Task items represent a task (an assigned, delegated, or self-imposed task to be performed within a timeframe).

Method	Field Name	Acceptable Values/Notes
<i>Get</i>	% Complete	Number (no decimals)
<i>Get/Set</i>	Actual Work	Time in Minutes
<i>Get/Set</i>	Billing Information	
<i>Get/Set</i>	Body	This is the “note” field
<i>Get/Set</i>	Categories	
<i>Get/Set</i>	Company	
<i>Get/Set</i>	Complete	True or False
<i>Get/Set</i>	Contacts	
<i>Get/Set</i>	Date Completed	Date
<i>Get/Set</i>	Due Date	Date
<i>Get/Set</i>	Importance	High, Normal, or Low.
<i>Get/Set</i>	Mileage	
<i>Get</i>	Owner	
<i>Get/Set</i>	Reminder	
<i>Get/Set</i>	Reminder Time	Timestamp
<i>Get</i>	Role	
<i>Get/Set</i>	Schedule + Priority	
<i>Get/Set</i>	Sensitivity	Confidential, Normal, Personal, Private
<i>Get/Set</i>	Start Date	Timestamp
<i>Get</i>	Status	Completed, Deferred, In Progress, Not Started, Waiting on Others
<i>Get/Set</i>	Subject	
<i>Get/Set</i>	Team Task	True or False
<i>Get/Set</i>	Total Work	Number

Note: The fields “% Complete”, “Date Complete”, “Complete”, “Status”, “Actual Work”, and “Total Work” recalculate their results based off of one another. If using these fields, we recommend exploring how each field relates to the other before adapting the fields to your solution’s use. For example:

- If “Status” field is set to Complete, then the “% Complete” field will be set to 100.
- If “% Complete” is set to 0, then the “Status” field will be set to “Not Started”.

Available Note Fields

Note items represent a simple note message for the user.

Method	Field Name	Acceptable Values/Notes
<i>Get/Set</i>	Body	This is the “note” field
<i>Get/Set</i>	Body Format	Plain Text, Rich Text, or HTML
<i>Get/Set</i>	Categories	
<i>Get/Set</i>	Color	Integer representing the colors. Possible values are: 0 for Blue, 1 for Green, 2 for Pink, 3 for Yellow, 4 for White.
<i>Get/Set</i>	Subject	
<i>Get/Set</i>	Width	Number value, in pixels
<i>Get/Set</i>	Height	Number value, in pixels
<i>Get/Set</i>	Left	Number value, in pixels
<i>Get/Set</i>	Top	Number value, in pixels

Note: The Width, Height, Left and Top fields adjust the dimensions and location of the note. These fields have no bearing on the content of the note, and are simply for controlling the appearance of the note when viewing it in a third-party application such as Microsoft Outlook or Apple Mail.

V. Contact Us

Successful integration of a FileMaker plug-in requires the creation of integration scripts within your FileMaker solution. A working knowledge of FileMaker Pro, especially in the areas of scripting and calculations, is necessary. If you need additional support for scripting, customization or setup (excluding registration) after reviewing the videos, documentation, FileMaker demo and sample scripts, then please contact us via the avenues listed below.

Phone: 760-510-1200

Email: support@productivecomputing.com

Please note assisting you with implementing this plug-in (excluding registration) is billable at our standard hourly rate. We bill on a time and materials basis billing only for the time in minutes it takes to assist you. We will be happy to create your integration scripts for you and can provide you with a free estimate if you fill out a Request For Quote (RFQ) at www.productivecomputing.com/rfq. We are ready to assist and look forward to hearing from you!

**This document was updated on November 15, 2023*